ΕΠΛ 232: Αλγόριθμοι και Πολυπλοκότητα

Κατ'οίκον Εργασία 2Α – Σκελετοί Λύσεων

1. Για τη σαφή διατύπωση του αλγόριθμου απαιτούνται τα εξής:
   - Διατήρηση της ροής που κτίζεται από τον αλγόριθμο.
   - Διατήρηση της περίσσειας χωρητικότητας για κάθε ακμή (δίκτυο περίσσεια).
   - Εύρεση ενός βραχύτερου επεκταμένου μονοπατιού στο δίκτυο περίσσειας.

   Για να πετύχουμε τα πιο πάνω σε υλοποίηση του δικτύου ροής με λίστα γειτνίασης χρησιμοποιούμε τις πιο κάτω δομές για αναπαράσταση του δικτύου ροής:

```
typedef struct node *link;

struct node {
     int v;                 /* το όνομα του κόμβου*/
     int cap;               /* χωρητικότητα της ακμής*/
     int flow;              /* ροή κατά μήκος της ακμής –
                               αρχικά  0*/
     int resflow;           /* υπόλοιπο χωρητικότητας κατά μήκος
                               της ακμής – αρχικά ίσο με το cap*/
     link back;             /* δείκτης προς την ακμή με την
                               αντίθετη κατεύθυνση*/
     link next;             /* δείκτης προς τον επόμενο κόμβο*/
};

struct fgraph{
     link list[V];
   };
```

   Επίσης χρησιμοποιούμε την πιο κάτω δομή για εύρεση και αναπαράσταση επεκταμένων μονοπατιών:

```
struct pathnode{
     int prev;
     int visited;
   };
```

   Η διαδικασία εύρεσης βραχύτερου επεκταμένου μονοπατιού γίνεται ως εξής (τα s και t είναι η πηγή και ο προορισμός του δικτύου, αντίστοιχα). Υποθέτουμε πως έχουμε στη διάθεσή μας υλοποίησης ουράς με τις πράξεις IsEmpty, Enqueue, Dequeue.

```
FindPath(fgraph G, int s, int t)

   pathnode path[V]= [(-1,0),…(-1,0)];
   queue Q;

   Enqueue(Q, s);
   path[s]->prev = 0;
   path[s]->visited = 0;
```

```
while (!IsEmpty(Q))
    u = Dequeue(Q);

    for (p = G->list[u]; p!= NULL; p = p->next)
            if (path[p->name]->visited == 0
                        AND path[p->name]->resflow > 0;
                  path[p->name]->visited == 1;
                  path[p->name]->prev == u;
                  if (p->name == t) return &path;
    return NULL;
}
```

Δοθέντος ενός μονοπατιού μπορούμε να υπολογίσουμε τη μέγιστη ποσότητας ροής που μπορεί να περάσει από το μονοπάτι με την πιο κάτω διαδικασία. Παρατηρούμε πως η διαδικασία ξεκινώντας από τον προορισμό διαπερνά το μονοπάτι προς τα πίσω υπολογίζοντας την ελάχιστη περίσσεια χωρητικότητα των ακμών του.

```
FindMinFlow (fgraph G, int s, int t, pathnode path[V])
    c = ∞;
    for (u = t; u != s; u = path[u].prev)
            v = path[u]->prev;

            for (p = G->list[v]; p->name !=u; p = p-> next);

            c = min (c, p->resflow);
```

Τέλος, δοθέντος ενός επεκταμένου μονοπατιού και της ποσότητας ροής που μπορεί να περάσει από αυτό μπορούμε ενισχύσουμε τη ροή σε ένα δίκτυο ροής με αυτή την ποσότητα ροής στο μονοπάτι ανανεώνοντας κατάλληλα τις πληροφορίες σχετικά με τις περίσσειες χωρητικότητας των ακμών όπως φαίνεται πιο κάτω. Σημαντικό σημείο στη συνάρτηση αποτελεί το γεγονός πως ενίσχυση μιας ροής είναι δυνατό να έχει ως αποτέλεσμα τη δημιουργία νέος ακμών (οπισθοδρόμησης) οι οποίες φέρουν αρνητική ποσότητα ροής και ανήκουν στο δίκτυο περίσσειας. Το σημείο αυτό παρουσιάζεται σκιασμένο στον κώδικα.

```
ExpandFlow (fgraph G, int s, int t, pathnode path[V], int c)
    for (u = t; u != s; u = path[u]->prev)
            v = path[u]->prev;
            for (p = G->list[v]; p->name !=u; p = p-> next)
            p->flow = c + p->flow;
            p->res = p->cap - p->flow;
            q = p->back;
            if q == NULL
                  q = Malloc(node);
                  q->next = G->list[u];
                  G->list[u] = q;
                  q->name = v;
                  q->cap = 0;
                  q->back = p;
                  p->back = q;
            q->flow = - p->flow;
            q->res = q->cap - q->flow;
```

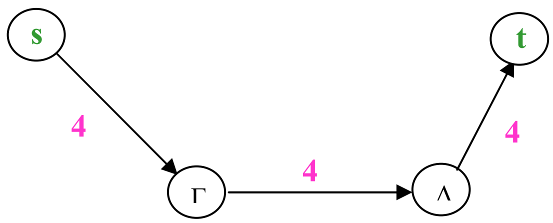
Η τελική διαδικασία έχει ως εξής:

```
EdmondsKarp(graph G, int s, int t)
path = FindPath(G, s, t);
    while (path != NULL)
          c = FindMinFlow(G, s, t, path);
          ExpandMinFlow(G, s, t, path, c);
```

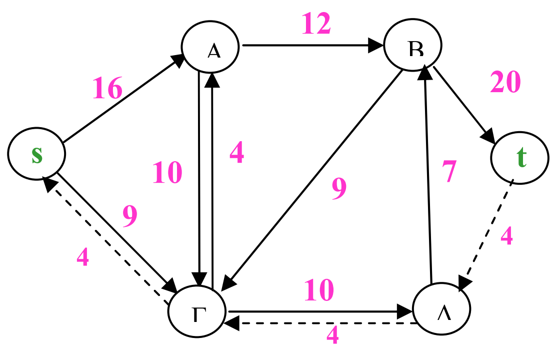
(β) Αρχική ροή 0, και δίκτυο περίσσειας ως προς την αρχική ροή το αρχικό δίκτυο ροής.

Επιλέγουμε το μονοπάτι s→Γ→Δ→t.

(Θετική) Ροή:

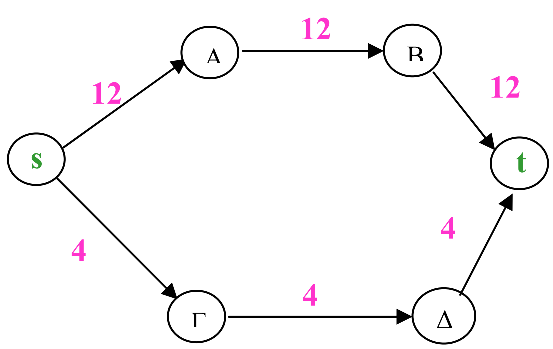


Δίκτυο περίσσειας:

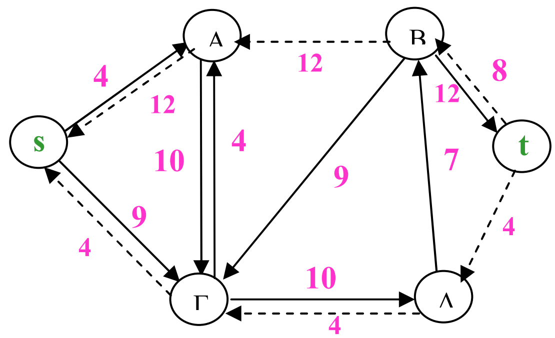


Επιλέγουμε το μονοπάτι s→A→B→t.

(Θετική) Ροή:

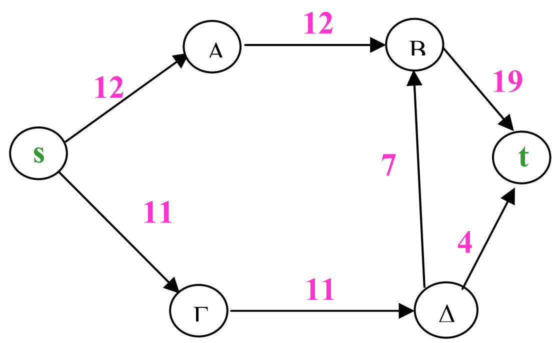


Δίκτυο περίσσειας:

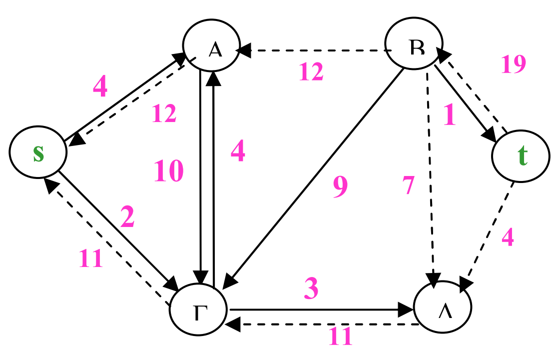


Επιλέγουμε το μονοπάτι s→ Γ→ Δ→ B→t.

(Θετική) Ροή:



Δίκτυο περίσσειας:



Δεν υπάρχει άλλο επεκταμένο μονοπάτι.

2. Έστω γράφος G = (V, E) και u,v ∈V. Ορίζουμε G_{uv} ως το δίκτυο ροής που αποτελείται από όλες τις ακμές E εκτός από όσες φθάνουν στην κορυφή u και όσες φεύγουν από την κορυφή v, πηγή τη u, προορισμό τη v και χωρητικότητα c(e) = 1, για κάθε ακμή e. Θέτοντας όλες τις χωρητικότητες ίσες με 1 διασφαλίζουμε ότι ο αριθμός των ακμών κατά μήκος της τομής είναι ίσος με τη χωρητικότητα κατά μήκος της τομής. Έστω f_{uv} μια μέγιστη ροή πάνω στο δίκτυο G_{uv}.

Ο ζητούμενος αλγόριθμος έχει ως εξής:

```
Διάλεξε κάποια κορυφή v ∈V
Τρέξε τον αλγόριθμο μέγιστης ροής
     σε όλα τα δίκτυα G_vu για κάθε u∈V-{v}.
Επέστρεψε την τιμή
          min_{u∈V-{v}} |f_uv|
```

<u>Λήμμα</u>

Ο βαθμός συνεκτικότητας, ΒΣ, ενός γράφου ισούται με $\min_{u \in V - \{v\}}|f_{uv}|$.

<u>Απόδειξη:</u>

Αρχικά θα δείξουμε ότι ΒΣ $\geq \min_{u \in V - \{v\}}|f_{uv}|$.
Παρατηρούμε ότι η τιμή $|f_{uv}|$ ουσιαστικά συμβολίζει τον αριθμό μονοπατιών που υπάρχουν ανάμεσα στους κόμβους u και v. Θεωρήστε την τομή με την ελάχιστη χωρητικότητα κατά μήκος τομής του δικτύου. Αυτή αποτελείται από $|f_{uv}|$ ακμές. Επομένως είναι δυνατό να αφαιρέσουμε $|f_{uv}| - 1$ ακμές της τομής αυτής από το δίκτυο ροής G_{uv} χωρίς να αποσυνδέσουμε τους κόμβους u και v. Το ίδιο ισχύει για όλα τα ζεύγη κορυφών u και v. Συνεπώς είναι δυνατόν να αφαιρέσουμε $\min_{u \in V - \{v\}} f_{uv} - 1$ ακμές από ένα γράφο χωρίς να τον αποσυνδέσουμε. Αυτό έπεται ότι ΒΣ $> \min_{u \in V - \{v\}}|f_{uv}| - 1$.

Στη συνέχεια θα δείξουμε το αντίθετο. Δηλαδή, ότι ΒΣ $\leq \min_{u \in V - \{v\}}|f_{uv}|$.
Έστω v ένας από τους κόμβους που ελαχιστοποιούν το $|f_{uv}|$. Από το θεώρημα μέγιστης ροής – ελάχιστης χωρητικότητας, υπάρχει μια τομή με χωρητικότητα $|f_{uv}|$. Αφού όλες οι χωρητικότητες είναι ακριβώς 1, ακριβώς $|f_{uv}|$ ακμές τέμνονται από την τομή. Αν αυτές οι ακμές αφαιρεθούν, δεν υπάρχει μονοπάτι από τον u στον v. Άρα ο γράφος αποσυνδέεται. Έτσι, το ζητούμενο έπεται.

Αυτό συμπληρώνει την απόδειξη του λήμματος.